CLEVER: a Cooperative and Cross-Layer Approach to Video Streaming in HetNets

Stefania Colonnese,[1] Francesca Cuomo,[1] Luca Chiaraviglio,[2] Valentina Salvatore,[1]
Tommaso Melodia,[3], Izhak Rubin,[4]
1) DIET Department, University of Rome Sapienza, Italy, email {name.surname}@uniroma1.it
2) EE Department, University of Rome Tor Vergata, Italy
3) Department of Electrical and Computer Engineering, Northeastern University, Boston, USA
4) Department of Electrical Engineering, University of California, Los Angeles, USA

Abstract—We investigate the problem of providing a video streaming service to mobile users in an heterogeneous cellular network composed of micro e-NodeBs (μeNBs) and macro e-NodeBs (MeNBs). More in detail, we target a cross-layer dynamic allocation of the bandwidth resources available over a set of μeNBs and one MeNB, with the goal of reducing the delay per chunk experienced by users. After optimally formulating the problem of minimizing the chunk delay, we detail the Cross LayEr Video stReaming (CLEVER) algorithm, to practically tackle it. CLEVER makes allocation decisions on the basis of information retrieved from the application layer as well as from lower layers. Results, obtained over two representative case studies, show that CLEVER is able to limit the chunk delay, while also reducing the amount of bandwidth reserved for offloaded users on the MeNB, as well as the number of offloaded users. In addition, we show that CLEVER performs clearly better than two selected reference algorithms, while being very close to a best bound. Finally, we show that our solution is able to achieve high fairness indexes and good levels of Quality of Experience (QoE).

Index Terms—heterogeneous cellular systems, small cells, video streaming, bandwidth allocation

1 Introduction

An ever increasing number of mobile users are using video streaming services provided through the cellular network infrastructure. The resource-intensive nature of video streaming, coupled with the forecasted increase in the number of connected devices, is forcing operators to evolve their networks towards the 5G paradigm [2]. It is anticipated that future 5G architectures will satisfy extremely high bandwidth demands, coupled with significant reductions in end-to-end latency.

In spite of the foreseen increase in spectral efficiency, 5G networks will still be faced with the problem of spectrum crunch caused by the scarcity of radio frequency spectra allocated for cellular communications. To solve this issue, a number of proposals have envisaged the use of heterogeneous networks (HetNets) [3], [4]. Such networks are composed of different tiers of cellular devices, with macro cells spread over the territory to provide basic connectivity, and small cells covering hot spot zones, i.e., areas where users (and traffic) tend to concentrate.

In this context, future HetNets will require flexible and dynamic use of all available resources. To meet 5G requirements in terms of bandwidth and delay, it is expected that small and macro cells will be jointly controlled [5]. This in turn requires to coordinate the management of traffic processes and spectral resources across cells, including the need for flexible design of the control and user planes [6].

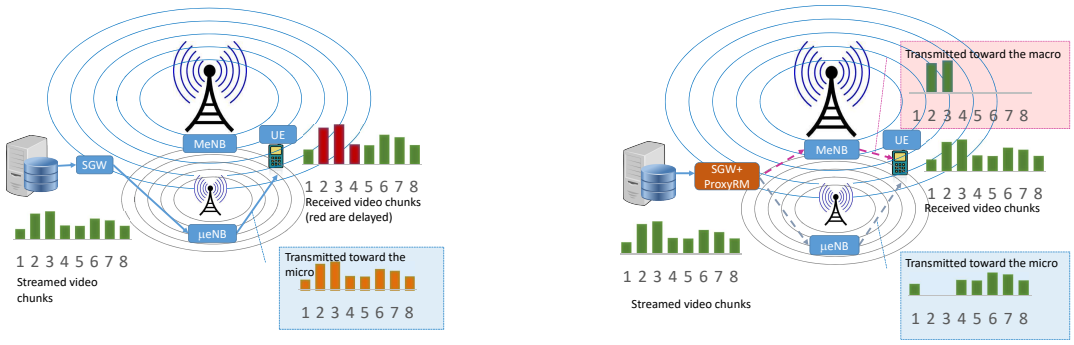
In this paper, we aim to answer the following questions: Is it possible to design cross-layer resource allocation mechanisms for HetNets that jointly control resources of the macro and small cells? What is the impact of such mechanisms on the delivery of resource-intensive services like video streaming? What are the potential benefits of developing flexible resource management approaches that allocate network resources based on application and lower layer based information? Is it possible to control the maximum number of users that need to be moved from a highly loaded cell to another one while ensuring Quality of Experience (QoE)? To provide answers to these compelling questions, we consider a network layout composed of small cells (named in the following *micro* cells) and a single macro cell. We focus on a 4G scenario (but the framework is valid also in 5G) where cells are served by eNode-Bs (eNBs). In this context, we present and study a solution, called Cross LayEr Video stReaming (CLEVER), to manage micro and macro eNBs bandwidth resources jointly together, while satisfying the users QoE (and the related video segment latency).

Our key contributions are as follows:

1) We provide the optimal formulation of the problem of minimizing the total experienced delay of video segments streamed in a system architecture composed by a macro cell and a set of micro cells; we demonstrate that the problem falls in the class of NP-Hard ones.
2) We derive CLEVER, a cross-layer resource allocation algorithm, used by HetNets to offload a subset of users from a micro cell to the macro cell when the spectral resources available at the micro cell are insufficient. Decisions made by the CLEVER scheme are based on information retrieved from the applica-

A preliminary version of this work appeared in [1].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2774298, IEEE Transactions on Mobile Computing



(a) Standard allocation and transmission toward the micro eNB (b) Proxy and Resource Manager supported allocation and transmission toward the micro and macro eNBs

Fig. 1. Two different architectural and resource allocation models.

tion layer (i.e., the video flow) as well as from status data obtained from lower layer (physical/data-link) entities.

3) We consider a general HetNet scenario as well as a layout that is based on a specific topological realization matching a real city.
4) We extensively assess the performance of CLEVER over the considered scenarios under a set of key performance metrics, including: the chunk delay, the amount of used resources on the eNBs, different fairness indexes, and the inter-stall time period.
5) We compare the performance of CLEVER against a set of reference algorithms, including: i) the case in which users are not offloaded, called NO OFFLOADING, ii) a solution to provide a best bound, referred as BEST BOUND, and iii) a greedy algorithm to solve the optimal problem, called GAGAP.

Our results show that a joint control of resources is highly beneficial when aiming to provide high quality video streaming services. We demonstrate that when making use of even a moderate portion of the macro bandwidth (typically ranging between 15% and 25% of the total bandwidth) one can provide a highly enhanced service to users that have been offloaded from micro cells. Moreover, CLEVER outperforms both the NO OFFLOADING and the GAGAP solutions, while being always very close to the BEST BOUND.

We thoroughly demonstrate the effectiveness of our proposed approach for the management of video streaming operations in a Hetnet, under QoE objectives. Our techniques also highlight approaches for intelligent cross-layer management for future 5G systems; e.g., providing methods for calculating the desired number of users to be transferred among micro and macro cells in connection with the realization of video streaming at specified QoE levels.

The rest of the paper is organized as following. An overview of the system architecture is reported in Sec. 2. The cross-layer resource allocation is detailed in Sec. 3. The considered scenarios are reported in Sec. 4. Performance evaluation results are presented and discussed in Sec. 5. Sec. 6 reviews the related work. Finally, conclusions are drawn in Sec. 7.

2 System Architecture and Assumptions

We consider an HTTP Adaptive Streaming (HAS) architecture where, at the server side, multiple versions of source video contents are stored. Each version is encoded for reception at a prescribed targeted video quality level, and is accordingly characterized by its specific flow bit rate. Previous works (e.g., [7], [8]) have shown that by using resource allocation schemes that are executed by a manger that resides at the cellular eNB side or at a central Resource Manager (RM), one can provide effective QoE support of video streaming at the video users while simultaneously providing for efficient utilization of available wireless resources. In this work, we extend this approach by considering a heterogeneous cellular layout (HetNet) that includes micro and macro cells. In addition, we employ a central resource management and allocation scheme that dynamically regulates the allocation of resources to the zone's macro and micro cells for performing video streaming transmissions, as well as for executing joint multipoint coordinations over multiple cell sites [9].

Figure 1 illustrates the considered architecture where a macro e-NodeB (MeNB) and a micro e-NodeB (μeNB) serve users that request video streaming from a video server. Base station nodes are connected through the network to a Serving Gateway (SGW). Without loss of generality, we assume that the Mobility Management Entity (MME) is located at the SGW facility. For each video stream, the server stores different stream replicas that are set to reproduce at distinct quality levels. Each encoded bit stream is parsed into video segments, which are referred to as "chunks". Each chunk encompasses one or more Group of Pictures (GOPs), which are addressed by means of URLs that are made available to the client through the HTTP protocol [10]. A chunk represents a segment of the video that lasts for several seconds (i.e., typically in the range $2-5$ [s]). In Fig. 1 eight consecutive chunks are represented and their height is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2774298, IEEE Transactions on Mobile Computing

proportional to the chunk size (typically measured in bytes). The client (identified as User Equipment, UE) sends requests for subsequent chunks to the server so as to receive them at a reception data rate sufficiently high for the client to avoid playout buffer starvation and video play stall events.

Definition of a Cross-Layer Approach. Our goal is to define an optimized bandwidth allocation scheme, while meeting video quality objectives. For this purpose, we devise a management scheme that considers the characteristics of the videos played by the users as well as the time-varying characteristics of the communications channel. Our management scheme provides for the allocation of spectral resources. In the following we refer to *bandwidth* to indicate a spectrum band expressed in [Hz] and we assign different portions of the spectrum band for the transmission of video chunks to users served by the eNBs. This operation is noted to be similar to the one described in [11] for chunk scheduling purposes; however, it is performed at distinctly different levels. Let us observe that scheduling in [11] operates at the application layer at the client side, based on the data throughput estimated at the client; in our paper the proposed spectral resource allocation procedure acts at the RM, it relies on cross-layer information, and it is transparent to the client.

The algorithm presented in this paper is executed by the proxy RM, noted as the proxyRM in Fig. 1. We observe that the UE is often capable of receiving video chunks of a specific stream from multiple eNBs. Thereby, we assume that the UE is able to receive video chunks from at least one μeNB and one MeNB node.

In accordance with the scheme here presented, the RM coordinates the serving eNBs in such a way that, for each chunk, a user is directed to receive the underlying video chunk from the eNB that has sent the previous chunk or to receive it from another eNB (such as the MeNB). Although we do not define the signaling procedures for switching from an eNB to another, this approach is in line with what defined in the Long Term Evolution Advanced (LTE-A) system for the joint multipoint coordination function [9].

By properly dynamically managing (i.e., on a chunk by chunk basis) and by coordinating the spectral resource allocation process over the μeNBs and the MeNB, we show that the system's performance behavior can be dramatically enhanced. In order to give intuitions of the proposed approach, Fig. 1 illustrates a qualitative example. In this scenario, a total of 8 chunks are streamed from a server to a user. Under the classical approach (Fig. 1(a)), all chunks are delivered through a single eNB. This cell can be, for instance, the μeNB to which the UE is connected at the start of the streaming process based on its proximity and/or the detection of a favorable signal quality condition. Consider the case under which during the video streaming of user i the bandwidth resources available at this μeNB becomes highly limited. Such a situation can be induced by the following conditions: i) during the video streaming of user i other UEs connected to the same eNB start video services; ii) the video spectral resources requested by supported users vary in time due to video fluctuations and/or UEs mobility. For the illustrative scenario, chunk numbers 2-3-4 in Fig. 1(a) in case of use of only μeNB are delivered at a time delay that impacts in a negative manner the quality (QoE) of the video streaming reception process (inducing stalls or quality fluctuations). In turn, with our approach, video chunks composing the stream can be dynamically delivered across either the MeNB or a μeNB, as highlighted in Fig. 1(b) by the dashed paths. In this example, two chunks (in this case, the ones demanding the highest level of bandwidth) out of the 8 chunks of the video stream played out by the user are sent toward the MeNB while the others are sent toward the μeNB. Notice that by dynamically allocating spectral resources in the MeNB to chunks 2 and 3, it is possible to reduce the μeNB load. As a result, not only chunks 2 and 3 are timely delivered, but also chunks of other contemporary videos streamed in the same cell (not shown in Fig.1(b) for simplicity's sake) benefit from the load reduction, and will incur in smaller delays.

We notice that, in the considered architecture, the overall delay of a chunk comes from the joint effect of network delays, delays due to the backhaul of the cooperative eNBs, radio scheduling delays at the selected eNB, transmission delay to download chunks on the wireless link. In this work, we assume that the delay depending on the backhaul infrastructure does not change when the serving eNB is changed. Therefore, the delay that we measure in intrinsically dependent only on the wireless access part of the network, which is typically the bottleneck. Besides, as we will show in the next sections, the number of offloaded chunks is kept as low as possible, compatibly with the performance.

3 Cross-Layer Resource Allocation

The objective of CLEVER is to define and implement a resource allocation algorithm across the μeNBs and MeNBs to: i) limit the saturation of the μeNBs, ii) fully exploit the spectrum band of the μeNB while limiting the use of the MeNB spectrum band (which may be used to provide additional services apart from video ones), and iii) guarantee an adequate service for users, coupled with low chunk delays.

More formally, by using the notation in Tab. 1, we consider a set $\mathcal{M}$ of eNBs with cardinality $M+1$. A number of M μeNBs (numbered with $m = 0, 1, ..., M-1$) are located inside the coverage area of one MeNB. In addition, we assume a set of video streaming users $\mathcal{S}$ inside the coverage area. The total number of users is denoted by N. Moreover, we consider a set of chunks $\mathcal{K}$. The total number of chunks is denoted by K. The m-th μeNB serves a subset $\mathcal{S}_k^{(m)} \subset \mathcal{S}$ of video streaming users for every chunk index $k \in \mathcal{K}$. The number of video streaming users served by eNB m at chunk index k is denoted by $N_k^{(m)}$. The i-th user of the m-th eNB receives a streaming video characterized by a given average video encoding rate. By considering a fixed chunk duration, denoted with τ, and time intervals with the same duration τ, the i-th user requests at chunk index k a video chunk of size $\lambda_k^{(i)}, i = 0, \cdots N-1$, corresponding to a playout of duration τ.[1] To avoid depletion at the user's video buffer, the chunk download time interval should not exceed τ; the net rate needed at the application layer to meet this goal is equal to $r_k^{(i)} = \lambda_k^{(i)}/\tau$.

1. We assume that all videos residing at the proxy are made time synchronous on a chunk oriented basis time; i.e., all start at the beginning of a chunk time interval τ.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2774298, IEEE Transactions on Mobile Computing

TABLE 1
Main Notation.

Symbol	Definition
$\mathcal{M}$, $M+1$	Set and number of eNBs (M μeNBs and one MeNB)
$\mathcal{K}$, K	Set and number of video chunks
$\mathcal{S}$, N	Set and number of video streaming users in the scenario
$\mathcal{S}_k^{(m)}$, $N_k^{(m)}$	Set and number of video streaming users at chunk index k for m-th μeNB, $m = 0, ..., M-1$
$\lambda_k^{(i)}$ [bit]	Video chunk size for user i at time k
$c_k^{(m,i)}$	Experienced CQI for user i from eNB m ($m = 0, ..., M$) at chunk index k
$\eta(c_k^{(m,i)})$ [bps/Hz]	Spectral efficiency related to CQI $c_k^{(m,i)}$
$B_k^{(m,i)}$ [Hz]	Requested bandwidth from user i to eNB m ($m = 0, ..., M$) at chunk index k
$\tilde{B}_k^{(m,i)}$ [Hz]	Served bandwidth to user i from eNB m ($m = 0, ..., M$) at chunk index k
$\delta_k^{(m,i)}$ [s]	Experienced delay of user i for downloading the video from eNB m ($m = 0, ..., M$) at chunk index k
$\mathcal{O}_k^{(m)}$	Set of offloaded users from m-th μeNB at time k, $m = 0, ..., M-1$
B_{TO} [Hz]	Overall amount of bandwidth to be offloaded to the MeNB
B_{MTO} [Hz]	Amount of bandwidth on the MeNB reserved for offloading the μeNBs users
$B_{MAX}^{(m)}$ [Hz]	Available bandwidth from eNB m ($m = 0, ..., M$)
$x_k^{(m,i)}$	Binary variable: 1 if user i is assigned to eNB m at chunk index k, 0 otherwise

In turn, the bandwidth $B_k^{(m,i)}$ needed by the i-th user when it is connected to the m-th eNB for timely download of the k-th user video chunk depends on $r_k^{(i)}$ and on the location dependent user experienced Signal to Interference plus Noise Ratio (SINR). In 4G systems, like the one here considered, the SINR is represented by a Channel Quality Indicator (CQI) denoted as $c_k^{(m,i)}$. More formally, $B_k^{(m,i)}$ is computed as:

$$B_k^{(m,i)} = \frac{r_k^{(i)}}{\eta(c_k^{(m,i)})\alpha} = \frac{\lambda_k^{(i)}}{\eta(c_k^{(m,i)})\alpha\tau} \quad \text{[Hz]} \tag{1}$$

where the factor $\alpha \in (0,1)$, (formerly introduced in [12]),[2] accounts for overhead induced by retransmissions incurred at lower protocol layers and $\eta(c_k^{(m,i)})$ is the spectral efficiency related to CQI $c_k^{(m,i)}$.[3]
We notice that the bandwidth required for streaming to user i at a given chunk index k depends on the CQI reported to the associated managing base station node, so that the actual required bandwidth depends on whether the user is served by the μeNB or the MeNB.

If a user connected to an eNB is assigned a bandwidth $\tilde{B}_k^{(m,i)} \leq B_k^{(m,i)}$, its chunk delivery time is delayed by:

$$\delta_k^{(m,i)} = \max\left\{\left(B_k^{(m,i)}/\tilde{B}_k^{(m,i)} - 1\right)\tau, 0\right\} \quad \text{[s]} \tag{2}$$

2. Therein, we observe that the throughput rate provided at the application layer is a fraction of the actual wireless data link determined by the lower layer protocols overhead; e.g., if TCP and MAC layer retransmissions occupy up to ρ_{TCP} and ρ_{MAC} % of the physical layer capacity, respectively, we obtain $\alpha \approx (1-\rho_{TCP}) \times (1-\rho_{MAC})$.

3. This parameter is intended as the average of the CQI related spectral efficiency during a chunk duration τ.

and the user's video buffer depletes accordingly. This occurs when, due to intrinsic video fluctuations or due to channel quality variations, the overall bandwidth requested by the users connected to the m-th eNB $\sum_{i=0}^{N_k^{(m)}-1} B_k^{(m,i)}$ exceeds the the m-th eNB available bandwidth, which is denoted by $B_{MAX}^{(m)}$.

3.1 Optimal Formulation

We then consider the optimal formulation of the problem, whose aim is to minimize the total delay experienced by users for each chunk index $k \in \mathcal{K}$. Specifically, we introduce the binary variable $x_k^{(m,i)}$, which takes value 1 if the i-th user it connected to the m-th eNB for chunk k, 0 otherwise. We assume that each user is connected to one eNB for each chunk index. More formally, we have:

$$\sum_{m\in\mathcal{M}} x_k^{(m,i)} = 1 \quad \forall i \in \mathcal{S}, \forall k \in \mathcal{K} \tag{3}$$

We then store in the continuous variable $\tilde{B}_k^{(m,i)} \geq 0$ the amount of bandwidth assigned to the i-th user from the m-th eNB at chunk index k. Clearly, $\tilde{B}_k^{(m,i)}$ is at most equal to the amount of bandwidth requested by the user, i.e., $B_k^{(m,i)}$. This constraint is expressed as:

$$\tilde{B}_k^{(m,i)} \leq B_k^{(m,i)} x_k^{(m,i)} \quad \forall m \in \mathcal{M}, \forall k \in \mathcal{K} \tag{4}$$

The total bandwidth assigned to the users has to be lower than the maximum amount of bandwidth available at the eNB. More formally, we have:

$$\sum_{i\in\mathcal{S}} \tilde{B}_k^{(m,i)} x_k^{(m,i)} \leq B_{MAX}^{(m)} \quad \forall m \in \mathcal{M}, \forall k \in \mathcal{K} \tag{5}$$

In the following, we store in the variable $\delta_k^{(m,i)} \geq 0$ the delay experienced by user i connected to eNB m at chunk index k:[4]

$$\delta_k^{(m,i)} \geq \left(B_k^{(m,i)}/\tilde{B}_k^{(m,i)} - 1\right)\tau \quad \forall m \in \mathcal{M}, i \in \mathcal{S}, \forall k \in \mathcal{K} \tag{6}$$

The MINIMUM TOTAL DELAY (MTD) problem is then formulated as follows:

$$\min \sum_{i\in\mathcal{S}} \sum_{m\in\mathcal{M}} \delta_k^{(m,i)} x_k^{(m,i)} \quad \forall k \in \mathcal{K} \tag{7}$$

subject to:

$$\begin{array}{ll} \text{Service constraint:} & \text{Eq. } (3) \\ \text{Bandwidth constraints:} & \text{Eq. } (4) - \text{Eq. } (5) \\ \text{Delay constraint:} & \text{Eq. } (6) \end{array} \tag{8}$$

under control variables: $x_k^{(m,i)} \in \{0,1\}$, $\tilde{B}_k^{(m,i)} \geq 0$.

Theorem 1. *The* MTD *problem falls in the class of NP-Hard problems.*

Proof. We consider a sub-case of the MTD problem, where the amount of bandwidth $\tilde{B}_k^{(m,i)}$ is preliminary assigned. As a result, $\delta_k^{(m,i)}$ becomes an input parameter, which is set equal to $\delta_k^{(m,i)} = \max\left[\left(B_k^{(m,i)}/\tilde{B}_k^{(m,i)} - 1\right]\tau, 0\right) \quad \forall m \in$

4. When the problem is optimally solved, a small ϵ should be added to $\tilde{B}_k^{(m,i)}$ to avoid an infinite delay setting.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2774298, IEEE Transactions on Mobile Computing

$\mathcal{M}, \forall i \in \mathcal{S}, \forall k \in \mathcal{K}$.[5] The problem can be formulated as follows:

$$\min \sum_{i \in \mathcal{S}} \sum_{m \in \mathcal{M}} \delta_k^{(m,i)} x_k^{(m,i)} \quad \forall k \in \mathcal{K} \tag{9}$$

subject to:

$$\begin{array}{ll} \sum_{i \in \mathcal{S}} \tilde{B}_k^{(m,i)} x_k^{(m,i)} \leq B_{MAX}^{(m)} & \forall m \in \mathcal{M}, \forall k \in \mathcal{K} \\ \sum_{m \in \mathcal{M}} x_k^{(m,i)} = 1 & \forall i \in \mathcal{S}, \forall k \in \mathcal{K} \end{array} \tag{10}$$

under control variables: $x_k^{(m,i)} \in \{0,1\}$. The previous formulation is known as the GENERALIZED ASSIGNMENT PROBLEM (GAP) [13]. Specifically, the GAP problem targets the minimization of the assignment costs of tasks to agents, under the constraints that the resources consumed by the tasks on the agents are limited, and that each task is assigned to an agent. In our case, the tasks are the users and the agents the eNBs. Moreover, the resources are the bandwidths assigned to the users, the assignment costs are the experienced delays, the maximum resource utilizations are the eNBs available bandwidths. Finally, the decision variables $x_k^{(m,i)}$ are the associations of users to eNBs. Since the GAP problem is NP-Hard [13], and it is included in the MTD problem, we can conclude that the latter is also NP-Hard.

□

Since the MTD formulation is very challenging to be solved even for small instances, we describe in the next subsection the CLEVER algorithm to practically tackle the problem.

3.2 CLEVER Algorithm

The main intuition of CLEVER is the capability of offloading a subset of video streaming users from a μeNB to an MeNB when the bandwidth on the μeNB is saturated. However, our solution targets also the reduction of μeNBs saturation events, by providing an efficient allocation of spectrum resources across the system's μeNBs and MeNBs nodes. In this context, we investigate the advantages of simultaneously offloading chunk download requests produced by a set of users to the region's MeNB. For this purpose, we consider the following offloading criteria. When a saturation event is detected, the most straightforward resolution approach is to select the users whose transmissions will be offloaded on the basis of data collected from physical/data-link protocol layer status states. One possibility would be then to move (i.e., re-associate) users based on their reported CQI values. This approach however does not take into account the intrinsic variability [14] of the size of video chunks over time, which may consequently result in large inefficiencies (and service unfairness among users). To solve this issue, we propose in this paper an offloading technique that is based on the state information derived from the application layer entity in addition to using physical/data-link protocol layer status data.

Our proposed algorithm operates on a chunk by chunk basis, and offloading is applied starting from the users' chunks that correspond to the smallest spectrum bandwidth request from their associated μeNB. Specifically, users chunks are selected for offloading on the basis of the bandwidth levels $B_k^{(m,i)}$ that they require, and are considered for transfer in increasing requested $B_k^{(m,i)}$ order as follows. The offloading set $\mathcal{O}_k^{(m)}$, $m = 0, .., (M-1)$ includes the minimum number of users' chunks, selected in increasing bandwidth order, such that either $\sum_{\mathcal{S}_k^{(m)} \setminus \mathcal{O}_k^{(m)}} B_k^{(m,i)} \leq B_{MAX}^{(m)}$ or the cardinality of $\mathcal{O}_k^{(m)}$ equals a maximum number of offloaded users, denoted as $N_{MAX}^{(m)}$. Thus, the m-th μeNB attempts to offload as many users chunks as necessary to allow it to serve the remaining chunks in a timely manner.

When a user is offloaded from a μeNB to the MeNB, the bandwidth required for streaming its video chunks changes. As noted by us in (1), we observe that the bandwidth $B_k^{(m,i)}$ required by the $i-th$ user from its m-th μeNB is directly proportional to the amount of data to be transferred $\lambda_k^{(m,i)}/\alpha\tau$, and inversely proportional to the reported spectral efficiency level $\eta(c_k^{(m,i)})$. Therefore, when a prescribed data quantity is offloaded to the MeNB, the required bandwidth is scaled by a factor $\eta(c_k^{(m,i)})/\eta(c_k^{(M,i)})$ relative to that required from the μeNB, in recognizing the different CQI values reported by a user to its μeNB $c_k^{(m,i)}$ vs. that reported to the MeNB $c_k^{(M,i)}$. The overall bandwidth request level imposed on the MeNB, as generated by chunk offloading requests from the μeNBs, is then computed as:

$$B_{TO} = \sum_{m=0}^{M-1} \sum_{i \in \mathcal{O}_k^{(m)}} B_k^{(M,i)} = \sum_{m=0}^{M-1} \sum_{i \in \mathcal{O}_k^{(m)}} \frac{\eta(c_k^{(m,i)})}{\eta(c_k^{(M,i)})} \cdot B_k^{(m,i)} \tag{11}$$

This total bandwidth request, indicated as B_{TO}, is served by a portion of the MeNB bandwidth level, denoted henceforth as "macro bandwidth to offload" (B_{MTO}).

Algorithm 1 summarizes the CLEVER computation steps. Since the downlink communication quality may change over the offloading decision time, our scheme handles offloading per chunk and not per stream. Specifically, the CLEVER algorithm offloads users on a chunk by chunk basis so that at chunk index k, users are temporarily offloaded for the purpose of streaming chunk k. The CQIs $c_k^{(m,i)}$ reported by the users are used to derive the bandwidth levels allocated to download the requested chunks to users served across the eNBs.

Steps 1-4 are used to derive the bandwidth requests of all users from the μeNBs in accordance to the sizes of their current video chunks and based on their reported channel quality states $c_k^{(m,i)}$. Steps 5-12 are designed to identify the set of users that may be offloaded to the MeNB, under the maximum number of offloaded users $N_{MAX}^{(m)}$. Step 13 computes the bandwidth levels to be allocated to the users that remain in the μeNB while steps 14-24 provide the computation of the bandwidth levels to be allocated to offloaded users. Specifically, in steps 16-24 we employ the bandwidth allocation algorithm that is identified as Minimum Average Delay (MAD) [15]. The MAD algorithm allocates the B_{MTO} bandwidth of the MeNB to the set $\mathcal{O}_k^{TOT} = \bigcup_m \mathcal{O}_k^{(m)}$ of offloaded users requiring $B_k^{(M,i)}$ so as to minimize the

5. Also here the case of infinite delay when $\widetilde{B}_k^{(m,i)} = 0$ is prevented by adding a small ϵ to $\widetilde{B}_k^{(m,i)}$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2774298, IEEE Transactions on Mobile Computing

Algorithm 1 Pseudo-Code describing the CLEVER controller

Input: k, $\mathcal{S}_k^{(m)}$, $N_k^{(m)}$, $N_{MAX}^{(m)}$, $\lambda_k^{(m,i)}$, $c_k^{(m,i)}$, $\eta(c_k^{(m,i)})$, $B_{MAX}^{(m)}$, B_{MTO}
Output: $\tilde{B}_k^{(m,i)}\ \forall i \in \mathcal{S}_k^{(m)} \setminus \mathcal{O}_k^{(m)}$, $\tilde{B}_k^{(M,i)}\ \forall i \in \mathcal{O}_k^{(m)}$, $\mathcal{O}_k^{(m)}$

1: **for** $m = 0...(M-1)$ **do**
2: **for** $i = 1..N_k^{(m)}$ **do**
3: **compute** $B_k^{(m,i)} = \dfrac{\lambda_k^{(m,i)}}{\eta(c_k^{(m,i)})\alpha\tau}$
4: **end for**
5: **set** $\mathcal{O}_k^{(m)} = \emptyset$
6: **for** $l = 1..N_{MAX}^{(m)}$ **do**
7: **if** $\sum_{i \in \mathcal{S}_k^{(m)} \setminus \mathcal{O}_k^{(m)}} B_k^{(m,i)} > B_{MAX}^{(m)}$ **then**
8: **pick** $i : B_k^{(m,i)} = \min_i \left[B_k^{\left(m, i \in \mathcal{S}_k^{(m)} \setminus \mathcal{O}_k^{(m)}\right)} \right]$
9: **update** $\mathcal{O}_k^{(m)} = \mathcal{O}_k^{(m)} \bigcup \{i\}$
10: **compute** $B_k^{(M,i)} = \dfrac{\eta(c_k^{(m,i)})}{\eta(c_k^{(M,i)})} \cdot B_k^{(m,i)}$
11: **end if**
12: **end for**
13: **compute** $\tilde{B}_k^{(m,i)} = B_k^{(m,i)}, i \in \mathcal{S}_k^{(m)} \setminus \mathcal{O}_k^{(m)}$
14: **end for**
15: **MAD** to **compute** $\tilde{B}_k^{(M,i)}$:
16: **define** $\Gamma = B_{MTO}$, $\Omega = \bigcup_m \mathcal{O}_k^{(m)}$,
17: $\mathcal{T} = \{i \in \bigcup_m \mathcal{O}_k^{(m)}, i \ s.t \ B_k^{(M,i)} \leq B_{MTO} \cdot \sqrt{B}_k^{(M,i)} / \sum_{l \in \bigcup_m \mathcal{O}_k^{(m)}} \sqrt{B}_k^{(M,l)}\}$
18: **while** $\mathcal{T} \neq \emptyset$ **do**
19: $\tilde{B}_k^{(M,i)} = B_k^{(M,i)}, i \in \mathcal{T}$;
20: $\Gamma = \Gamma - \sum_{i \in \mathcal{T}} \tilde{B}_k^{(M,i)}$;
21: $\Omega = \Omega \setminus \mathcal{T}$
22: $\mathcal{T} = \{i \in \Omega, i \ s.t \ B_k^{(M,i)} \leq \Gamma \cdot \sqrt{B}_k^{(M,i)} / \sum_{l \in \Omega} \sqrt{B}_k^{(M,l)}\}$
23: **end while**
24: $\tilde{B}_k^{(M,i)} = \Gamma \cdot \sqrt{B}_k^{(M,i)} / \sum_{l \in \Omega} \sqrt{B}_k^{(M,l)}, i \in \Omega$;

experienced average delay. This is achieved by the following allocation formula:

$$\tilde{B}_k^{(M,i)} = \min \left(B_k^{(M,i)}, \varphi \frac{\sqrt{B_k^{(M,i)}}}{\sum_{l \in \mathcal{O}_k^{TOT}} \sqrt{B_k^{(M,l)}}} \cdot B_{MTO} \right) \quad (12)$$

where the constant φ is such that $\sum_{l \in \mathcal{O}_k^{TOT}} \tilde{B}_k^{(M,l)} = B_{MTO}$. Let us notice that the implicit definition of φ appearing in Eq. (12) is numerically implemented by means of the recursive assignment in steps 16-24. The calculated bandwidth values act as constraints imposed on MAC layer schedulers.

A few remarks are in order. The CLEVER algorithm selects users for offloading their chunks at time k based on the bandwidth that they require from the μeNB. The rational behind this choice is that i) the offloading granularity is the smallest, resulting in the least load on the MeNB, and ii) the offloading fairness is high, since the offloading set selection is randomized by the intrinsic variability of the chunk size, instead of being related to the channel quality which may be very stable over time. Nevertheless, the proposed offloading criterion may be generalized to account for different parameters, e.g., the macro-related channel quality as well as for the actually available macro bandwidth; this issue is left for further study. Eventually, we observe that the CLEVER allocation algorithm perfectly suites HAS because it proactively handles undesired rate fluctuations by adapting the users spectrum allocated bandwidth so as to reduce the risk of unnecessary and annoying quality fluctuations. Moreover, our solution does not interfere with client initiated quality switching when, despite of the bandwidth allocation strategy, a change of rate is definitely needed to account for the actual channel capacity.

3.2.1 Computational Complexity

We first discuss the time complexity of our solution. Focusing on time complexity, CLEVER initially computes the requested bandwidth $B_k^{(m,i)}$ for each user and each eNB (lines 1-4 of Alg. 1). The complexity of this operation is $O(M \times N)$. Then, users may be offloaded to the MeNB (lines 5-12). This operation, repeated for all the μeNBs, has a complexity of $O(M \times N^2)$. Eventually, the amount of served bandwidth is set for the users that are not offloaded (line 13), resulting in a complexity $O(M \times N)$. In the final part (lines 15-24), the MAD routine is run for the users that are offloaded to the MeNB. Since the φ parameter appearing in Eq. 12 has to be computed with a recursive routine over the users set (lines 18-23), the resulting complexity is $O(N!)$. Even though this complexity may appear very high, we point out that, in practical scenarios, it can be kept pretty limited, due to the following reasons: firstly, the condition $\mathcal{T} \neq \emptyset$, triggering the recursive routine, correspond to few configurations of the requested bandwidth set (5% out of the total runs in our simulations); secondly, even when the condition applies, the while cycle is mostly iterated just once. Finally, the amount of served bandwidth is also set for the users offloaded to the MeNB (line 23), with a complexity $O(N)$. As a result, the actual overall complexity of the CLEVER algorithm is approximated as $\approx O(M \times N^2)$.

We then consider the space complexity of CLEVER. Specifically, our solution requires to store the requested bandwidth $B_k^{(m,i)}$ and the assigned bandwidth $\widetilde{B}_k^{(m,i)}$ in two matrices, each of them requiring to store up most $(M+1) \times N$ elements for each chunk index k. In addition, the set of offloaded users is stored in $\mathcal{O}_k^{(m)}$, whose size is equal to $(M+1) \times N$ for each chunk index k. As a result, the overall space complexity of CLEVER is pretty limited.

3.2.2 Implementation Issues

As for the implementation of CLEVER in a real system, we notice that several papers propose to use the application layer information typically adopted by the video entity (client/server) at lower layers. The paper [16] proposes, like in our case, the use of multiple nodes (named helpers) that serve multiple wireless users over a given geographic coverage area in a dynamic adaptive video context. Also in their case they adopt a cross-layer approach where the information that needs to be exchanged between the layers is the length of the users request queues, together with the chunk requests. The benefits of cross layer approaches for video streaming are widely discussed in the paper [17] where the authors present the many advances that can be achieved by taking video-specific (i.e., application layer) information into account when making lower layer decisions. In all cases it is used a network agent, like the RM in our approach, having in charge the cross-layering orchestration. In our

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2774298, IEEE Transactions on Mobile Computing

CLEVER implementation we request only the knowledge, at a RM entity, of the chunk size (which is already available at the server side) and of the CQI perceived by the i-th user that is typically carried by the LTE uplink channels PUCCH (in the periodic CQI case) or PUSCH (in the aperiodic CQI case).

As for the application layer information (chunk size), this can be transferred by the server to the RM in the form of the media presentation description (MPD) file, which is an XML document containing information about media segments, their relationships and information necessary to choose between them, and other metadata that may be needed by clients [18].

As for the CQI, in our implementation we suppose to use the aperiodic CQI reports triggered by different eNBs that can support the uses (MenB and μeNBs). Since our resource allocation timing is in the order of the chunk durations (i.e., seconds) the aperiodic CQI trigger could have the same period of a chunk.[6] The aperiodic CQI report is transmitted on PUSCH, together with UL data or alone. An assessed protocol for implementing the coordination of BS to achieve a cooperative transmission in the LTE-A downlink is the CoMP [19]. Finally, in our approach, neither the content provider nor the end user imply a knowledge of the fact that the CLEVER algorithm is used to allocate the wireless resources. Indeed, the CLEVER algorithm addresses the coordination of the eNBs and the radio access. On the contrary, the video server keeps the same information used in the conventional approach. Moreover, the end user transmits the PHY/MAC information typically sent to the eNB and poses HTTP GETs toward the network (via the indicated serving eNB) in a Dynamic Adaptive Streaming over HTTP (DASH) standard manner.

4 Considered Scenarios

We first describe the cellular scenario characteristics and then the adopted video traces.

4.1 Cellular Scenarios

We initially consider a general scenario and then we focus on an actual scenario involving a layout based on a location in an Italian city.

4.1.1 General Scenario

We take into account a service area covered by one MeNB and three μeNBs. The service area is modeled as a square of dimensions 5×5 [km^2]. Tab. 2 reports the main scenario parameters. Specifically, we assume that the MeNB covers the service area, while the μeNBs are placed in three hot spot zones, each of them covering a circle of small radius (equal to 50 [m]). Apart from the considered MeNB (which is placed in the center of the area), a tier of neighboring MeNBs surrounds the considered service area. In particular, by assuming a hexagonal layout and 120^o sectorization, we have placed six MeNBs to surround the central one. The distance between adjacent MeNBs is equal to 5 [km]. The positions of the μeNBs have been chosen such that each μeNB is at a distance of 1.5 [km] from the MeNB. Users are widely spread over the area of operations. More in detail, we focus on video users that are connected to the μeNBs, which may be eventually offloaded to the MeNB when the μeNB experiences bandwidth limitations. We assume a fixed number of video users $N_k^{(m)}$=16 to be initially connected to each μeNB. In the following, users are distributed in clusters centered on the μeNBs. More in detail, the coordinates of the users' spatial locations are i.i.d. Gaussian random variates of mean equal to the μeNB location and standard deviation equal to the μeNB radius, which is set equal to 50 [m]. Finally, a small number of users (which is set to 3 in our case) is randomly spread over the MeNB service area. Fig. 2 shows the central MeNB, the μeNB, and a realization of user positions.

Focusing then on the available bandwidth, the MeNBs are adopting a shared $B_{MeNB} = 20$ [MHz] bandwidth (as for instance in an LTE system). As a consequence, the central MeNB may potentially interfere with its surrounding MeNBs. Each μeNB is allocated a $B_{\mu eNB} = 5$ [MHz] bandwidth, at a separate band from that assigned to the MeNBs and shared with the other μeNBs. Consequently, transmissions by two distinct μeNBs may potentially inter-

6. Notice that during the chunk transmission, periodic CQI reports can continue with the serving eNB to allow dynamic PHY transmission adjustments.

TABLE 2
Scenario Parameters

Parameter	Value	
	General Scenario	Real City Scenario
Coverage Area MeNB	5 x 5 [km]^2	1.7834 [km]^2
μeNB Radius	50 [m]	50 [m]
Number of sectors per MeNB	3	3
Number of μeNBs	3	13
Number of users per μeNB ($N_k^{(m)}$)	16	50
μeNB distance range from MeNB	1500 [m]	187-1053 [m]
Receiver node power	-97.5 [dBm]	
Min receiver sensitivity	-107.5 [dBm]	
Maximum eNB TX power	36 [dBm] ($P_{\mu eNB}$), 43 [dBm] (P_{MeNB})	
eNB Antenna Gain G	10 [dB] (μeNB), 13 [dB] (MeNB)	
eNB Operating Frequency	2 [GHz]	
eNB Total Bandwidth	5 [MHz] ($B_{\mu eNB}$) 20 [MHz] per sector (B_{MeNB})	

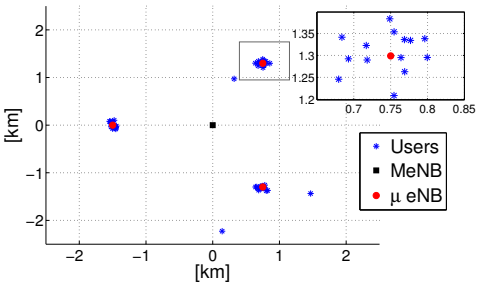


Fig. 2. General scenario: central MeNB, μeNB, and realization of users positions. The figure reports also an inset for a single μeNB.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2774298, IEEE Transactions on Mobile Computing

TABLE 3
Parameters for the Walfisch-Ikegami Model

Parameter	Value
eNB height	10 [m] (micro), 20 [m] (macro)
Average building height	20 [m]
Average road width	20 [m]
Average building separation	36 [m]
Receiver Height	1.5 [m]

TABLE 4
Downlink CQI c and net spectral efficiency $\eta(c)$ (Case of downlink Single Input Single Output system).

CQI index c	Net spectral efficiency $\eta(c)$ [bps/Hz]	CQI index c	Spectral efficiency $\eta(c)$ [bps/Hz]
0	out of range	8	1.3389
1	0.1065	9	1.6833
2	0.1640	10	1.9111
3	0.2638	11	2.3222
4	0.4211	12	2.7278
5	0.6139	13	3.1611
6	0.8222	14	3.5778
7	1.0333	15	3.8833

fere with each other. Finally, we assume that a portion equal to 4 [Mhz] of the B_{MeNB} is always reserved to the voice traffic.

Once the positions of the MeNBs, the μeNBs and the users have been generated, the downlink SINR experienced by each user is evaluated. To this end, we have used the statistical path propagation model of Walfish-Ikegami [20], [21] to compute the path loss. More in detail, this model takes into account the roof-top propagation in flat urban areas and it is therefore well suited to compute the urban transmission loss in our cellular mobile system. Without loss of generality, we assume Non-Line-of-Sight propagation in our experiments. The main parameters of the model are reported in Table 3. The SINR of the i-th user with respect to its μeNB $m = 0, ..., (M-1)$ is computed as:

$$SINR_{(m,i)} = \frac{G_{(m,i)} P_{\mu eNB}}{B_{\mu eNB} E_0 + \sum_{m'=0, m' \neq m}^{M-1} G_{(m',i)} P_{\mu eNB}} \quad (13)$$

where $E_0 = -174$ [dBm/Hz] is the power spectral density of the background thermal noise and $G_{(m,i)}$ is the path gain of the m-th μeNB to the i-th user. In a similar way, we have computed the SINR of the central MeNB to the i-th user, which is denoted as $SINR_{(M,i)}$. To this end, we have considered the MeNBs surrounding the central one as sources of interference.

The SINR level is then translated into a CQI level (by using the same procedure used in the analysis of LTE systems). The CQI is defined as an integer value that assumes values ranging between 1 and 15, with higher values corresponding to higher communications quality channels (and thus inducing higher SINR levels). We use the conversion described in [22] and [23], which is based on the following relationship:

$$CQI = \left[1 + \frac{7}{13} (SINR^{dB} + 6) \right], \quad -6 \leq SINR^{dB} \leq 20 \quad (14)$$

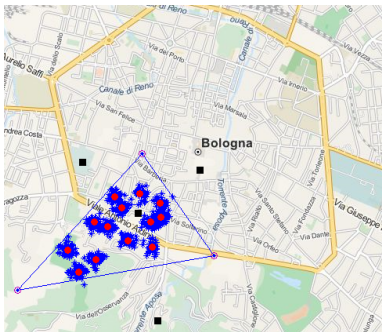


Fig. 3. Real city scenario: users, MeNBs and μeNBs.

TABLE 5
Video Parameters

Parameter	Value		
	quality q=2	quality q=3	quality q=4
Number of sequences	5	2	2
Encoding rates [Mbps]	$HP = 1.084$, $FN = 1.336$, $BP = 1.351$, $SP = 1.168$, $LH = 1.198$	$HP = 0.528$, $FN = 0.560$	$HP = 0.274$, $FN = 0.294$
PSNR [dB]	$HP = 44$, $FN = 44$, $BP = 45$, $SP = 44$, $LH = 45$	$HP = 42$, $FN = 42$	$HP = 40$, $FN = 40$
Duration	$K = 1200$ chunks (2400 [s] @ $\tau = 2$ [s])		

where the $[.]$ operator denotes the closest integer. Tab. 4 shows different CQI values and the corresponding realized net spectral efficiencies computed as in [7].

4.1.2 Real City Scenario

In this subsection, we present an actual scenario located in the city of Bologna, Italy. Specifically, our goal is to investigate the influence of a realistic cell deployment on the performance of the offloading scheme. Moreover, differently from the general scenario, the interference between the μeNBs is not negligible. Finally, we employ a system implementation that includes the use of a MIMO antenna system. To this aim, the spectral efficiency values shown in Tab. 4 are all multiplied by a gain factor that is equal to 2, in accordance to the results cited in [24].

Given the positions of the MeNBs reported in [25], we have focused on a neighborhood of the city covered by a single MeNB. The MeNB is then surrounded by a set of interfering MeNBs, as reported in Fig. 3. The area under consideration consists of the coverage area of the selected MeNB (i.e., the Voronoi area reported in the figure). Inside this area, we have then placed multiple μeNBs. More in depth, the μeNBs are placed at different points of interest, e.g., university buildings, commercial company buildings, and public squares. Finally, the users positions are distributed in clusters centered on the μeNBs, following the same procedure of the general scenario case.

4.2 Adopted Real Video Traces

We employ H.265/HEVC 10.1 encoded videos traffic traces publicly available from [26]. The considered video traces

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2774298, IEEE Transactions on Mobile Computing

represent the sequences of the encoded frame of the following movies: *Harry Potter* (HP), *Finding Neverland* (FN), *Lake House* (LH), *Speed* (SP), *black Planet* (BP). The traces have been recorded by [27], by encoding 1920x1080 spatial resolution videos at 24 frames per second, using a 24 frames long GoP. Given the frame size traces, we have built the set $\mathcal{K}$ of chunks. In our case, $K = 1200$ chunks and $\tau = 2$ [s], i.e., two GoPs per chunk. We have then considered different quantization levels (namely 25, 30 and 35) corresponding to the quality levels $q = 2, 3$ and 4. Moreover, the videos are characterized by different encoding bit rates and different Peak Signal to Noise Ratios (PSNRs), [7] which are summarized in Tab. 5.

5 Performance Evaluation

We then evaluate CLEVER over the different scenarios. To this aim, we have built a simulator based on the Matlab software. More in detail, our software starts from the generation of a random set of users around the eNBs, which are distributed according to the general or the realistic scenarios described above. In each simulation, we randomly associate one of the video traces in Tab. 5 to each user. In addition, we assume a random starting temporal index within the video chunk size sequence, which is cyclically shifted so that all the users' video streams have the same length. Next, we consider the chunk-by-chunk transmission of the video sequence. At each chunk, the net throughput rate provided at the application layer is a fraction $\alpha < 1$ of the actual wireless capacity, which in turn is determined by the user's assigned bandwidth and by the location dependent user's spectral efficiency (computed in accordance to the model described in Sec. 4). Let us notice that within each simulation run we inhibit the quality switching and we assume the initial buffering to be high enough to prevent rebuffering events for all the considered algorithms.[8] This allows us to fairly compare the delays produced by each allocation algorithm independently of the particular rate adaptation strategy and rebuffering procedure, since these latter may cause undesired rate oscillations and buffer-related randomic events affecting the stability of the numerical simulation results. Let us notice that even under the constant rate constraint, the video chunk sequence exhibit large fluctuation in size. In fact, as detailed in [7], the chunk size tends to follow a heavy tailed (e.g. Gamma) distribution, corresponding to a not negligible probability of occurrence of very large chunks. Hence, the throughput needed by video users also exhibits significant fluctuations in time, giving in turn rise to fluctuating bandwidth requests.

In order to better assess the CLEVER performance, we have also coded in our simulator a set of reference algorithms, named NO OFFLOADING, BEST BOUND and GAGAP [28]. More in detail, the NO OFFLOADING heuristic does not offload users to the MeNB, and the MAD routine is separately executed on the users connected to the μeNBs and on the ones connected to the MeNB. Focusing on the BEST BOUND, this solution operates on every μeNB and corresponds to applying the MAD routine on all the μeNB users, by considering an overall bandwidth equal to $B_{\mu eNB}$+B_{MTO} for each μeNB. The BEST BOUND is an upper bound for CLEVER under a twofold respect: firstly, it makes use of an undivided bandwidth, which achieves overall average delays lower or equal than the delays achieved by a separate handling of micro and offloaded users; secondly, in evaluating the bandwidth requests, it always uses the one requested to the μeNBs, which is lower or equal to the bandwidth that would be requested to the MeNB by offloaded users (due to the different CQIs experienced by users w.r.t the two kinds of eNBs). Appendix A reports a detailed description of this solution. In addition, GAGAP is a Greedy Algorithm for solving the Generalized Assignment Problem, which is adapted from [28] to our context. The GAGAP algorithm is a solution in which a user may be either: i) served with the amount of requested bandwidth, i.e., $\tilde{B}_k^{(m,i)} = B_k^{(m,i)}$, or ii) not served at all, i.e., $\tilde{B}_k^{(m,i)} = 0$. More in depth, GAGAP iteratively assigns and serves the users, until there is bandwidth available on the eNBs. Consequently, the delay $\delta_k^{(m,i)}$ is equal to 0 for the served users. However, in contrast to CLEVER, NO OFFLOADING and BEST BOUND solutions, with GAGAP a user may be not served by any eNB, i.e., when there is not enough bandwidth available on the eNBs. Therefore, the user that is not served experiences an outage. In general, this condition is more critical than introducing a delay. In this context, we denote with o_k^i a binary variable taking the value 1 if user i is in outage for chunk index k, 0 otherwise. We refer the reader to Appendix B for the detailed description of GAGAP.

TABLE 6
Available bandwidth for the users connected to the μeNB or offloaded to the MeNB for the different algorithms

Algorithm	$B_{MAX}^{(m)}$	
	$m = 0, .., (M-1)$	$m = M$
NO OFFLOADING	$B_{\mu eNB}$	-
BEST BOUND	$B_{\mu eNB}$+B_{MTO}	-
GAGAP	$B_{\mu eNB}$	B_{MTO}
CLEVER	$B_{\mu eNB}$	B_{MTO}

5.1 Performance Metrics

In order to assess the performance of the different allocation algorithms, we firstly consider the average chunk delay per user, which turns out into undesired video client's buffer depletion; secondly, we analyze a few metrics characterizing the bandwidth occupancy of the CLEVER algorithm and its competitors; finally, we consider the fairness of the algorithms both in terms of bandwidth occupancy as well as in terms of service quality metrics [29].

To elaborate, let us formally introduce the considered metrics. The average per chunk delay $\bar{\delta}$ is computed as:

$$\bar{\delta} = \frac{\sum_{k\in\mathcal{K}} \sum_{m\in\mathcal{M}} \sum_{i\in\mathcal{S}_k^{(m)}} \delta_k^{(m,i)} x_k^{(m,i)}}{K \cdot N} \qquad (15)$$

In addition, we consider the amount of bandwidth not used on the μeNBs and on the portion of the MeNB bandwidth reserved to offloaded users. This metric, normalized

7. The encoding PSNR is defined as $PSNR = 255^2/MSE$, being MSE the Mean Square value of the encoding Error.

8. In our simulation conditions this corresponds to assume 30 [s] buffering for the CLEVER algorithm with $B_{MTO} = 5$ [MHz].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2774298, IEEE Transactions on Mobile Computing

by the total available bandwidth and averaged over the different chunks, is denoted as fraction of not used bandwidth (F_{NUB}). F_{NUB} is formally expressed as:

$$F_{NUB} = \frac{\sum_{k\in\mathcal{K}}\sum_{m\in\mathcal{M}}\left(1-\frac{\sum_{i\in\mathcal{S}_k^{(m)}}\tilde{B}_k^{(m,i)}}{B_{MAX}^{(m)}}\right)}{K\cdot(M+1)} \quad (16)$$

where $B_{MAX}^{(m)}$ is set in accordance to Tab. 6 for the different algorithms.

Moreover, we denote with F_{SB} the fraction of served bandwidth over the requested one. F_{SB} is computed as:

$$F_{SB} = \frac{1}{K}\sum_{k\in\mathcal{K}}\frac{\sum_m\sum_{i\in\mathcal{S}_k^{(m)}}\tilde{B}_k^{(m,i)}x_k^{(m,i)}}{\sum_m\sum_{i\in\mathcal{S}_k^{(m)}}B_k^{(m,i)}x_k^{(m,i)}} \quad (17)$$

Apart from the aforementioned average metrics, we have also collected information about the algorithm fairness. To this aim, we have introduced the average fairness in the delay experienced by users J_D by exploiting the well-known Jain's fairness index:

$$J_D = \frac{1}{K}\sum_{k\in\mathcal{K}}\frac{\left(\sum_{m\in\mathcal{M}}\sum_{i\in\mathcal{S}_k^{(m)}}\delta_k^{(m,i)}x_k^{(m,i)}\right)^2}{N\cdot\sum_{m\in\mathcal{M}}\sum_{i\in\mathcal{S}_k^{(m)}}\left(\delta_k^{(m,i)}x_k^{(m,i)}\right)^2} \quad (18)$$

In a similar way, we introduce the average Jain's fairness index for the outage J_O (which is computed from the outage variables o_k^i of the GAGAP algorithm):

$$J_O = \frac{1}{K}\sum_{k}\frac{\left(\sum_i o_k^i\right)^2}{N\cdot\sum_i\left(o_k^i\right)^2} \quad (19)$$

and the average Jain's fairness index of the difference between the requested bandwidth and the served one $B_k^{(m,i)}-\tilde{B}_k^{(m,i)}$, which is denoted with J_{BD}:

$$J_{BD} = \frac{1}{K}\sum_{k\in\mathcal{K}}\frac{\left[\sum_m\sum_{i\in\mathcal{S}_k^{(m)}}\left(B_k^{(m,i)}-\tilde{B}_k^{(m,i)}\right)x_k^{(m,i)}\right]^2}{N\cdot\sum_m\sum_{i\in\mathcal{S}_k^{(m)}}\left[\left(B_k^{(m,i)}-\tilde{B}_k^{(m,i)}\right)x_k^{(m,i)}\right]^2} \quad (20)$$

5.2 Results from the general scenario

We then run the different algorithms over the general scenario. Unless otherwise specified, the results are obtained from 50 independent runs for generating the user positions and the requests of bandwidth. We initially compare the CLEVER algorithm with the NO OFFLOADING solution. Focusing on CLEVER, we initially set the maximum number of offloaded users equal to the number of users per μeNB, i.e., $N_{MAX}^{(m)} = N_k^{(m)}$. We then vary the B_{MTO} parameter, which governs the MeNB bandwidth used by CLEVER to offload the users (in accordance to Tab. 6). Fig. 4 reports the obtained Cumulative Distribution Functions (CDFs) of the average delay $\bar{\delta}$ obtained by the two solutions. When the NO OFFLOADING solution is applied, the delay experienced by users tends to be large, i.e., more than 0.035 [s] on average per chunk. On the other hand, when CLEVER is adopted, the delay is already more than halved when

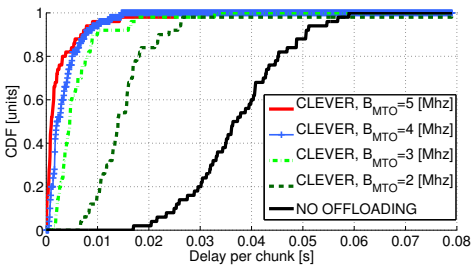


Fig. 4. CDF of the average delay per chunk $\bar{\delta}$ for CLEVER (with different values of B_{MTO}) and NO OFFLOADING solutions.

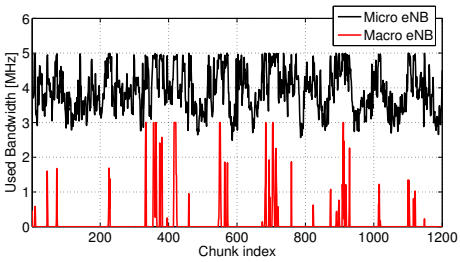


Fig. 5. Amount of used bandwidth with CLEVER on a single μeNB and on the B_{MTO} portion of the MeNB vs. the chunk index.

B_{MTO}=2 [Mhz], i.e., when 10% of the MeNB bandwidth is reserved for the offloading of the users. Eventually, the delay further decreases when B_{MTO} is increased, due to the fact that a larger portion of MeNB bandwidth is made available to the offloaded users.

In the following, we investigate the effectiveness of CLEVER in managing the amount of used bandwidth. More in detail, we consider the amount of bandwidth used on a μeNB and the one used by the offloaded users on the corresponding sector of the MeNB. Fig. 5 plots the obtained results vs. the chunk index, as an outcome of a single run with $B_{MTO} = 3$ [Mhz]. Several considerations hold in this case. First, the bandwidth used on the μeNB is naturally always lower than $B_{\mu eNB}$. Second, the MeNB is efficiently exploited to offload users, which may even require the whole B_{MTO}. Third, the amount of used bandwidth on both the μeNB and the MeNB sector tends to notably vary over time, as a consequence of the variation of the requested bandwidth from users.

In order to better assess the benefits introduced by CLEVER, we compare our solution with the BEST BOUND. Fig. 6 reports the delay per chunk of CLEVER and BEST BOUND for different values of B_{MTO}. Bars report average values, while error bars report the confidence intervals (computed with a confidence level of 95%). In this case, when B_{MTO} is increased, the delay attained by CLEVER tends to be reduced, being very close to the BEST BOUND when $B_{MTO} = 5$ [Mhz]. This corresponds to the case in which the percentage of bandwidth on the MeNB reserved for offloading is equal to 25% of the total one. On the other hand, when B_{MTO} decreases, the delay of CLEVER increases w.r.t. also to the BEST BOUND. However, we point out that the BEST BOUND is a pretty optimistic solution, due to the fact that: i) the μeNB is always able to exploit a total of $B_{\mu eNB} + B_{MTO}$ available bandwidth

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2774298, IEEE Transactions on Mobile Computing

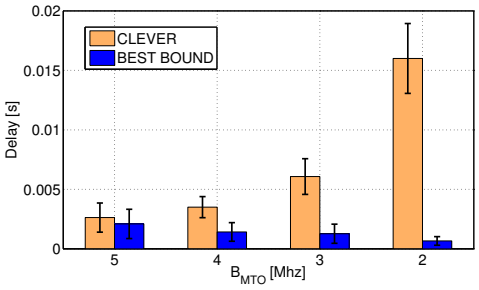


Fig. 6. Delay per chunk vs. B_{MTO} for CLEVER and BEST BOUND.

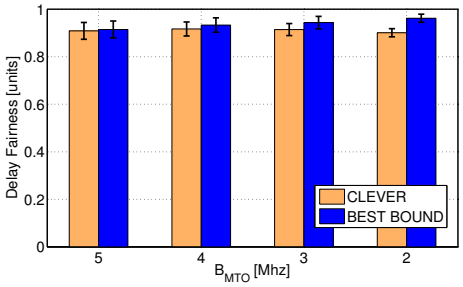
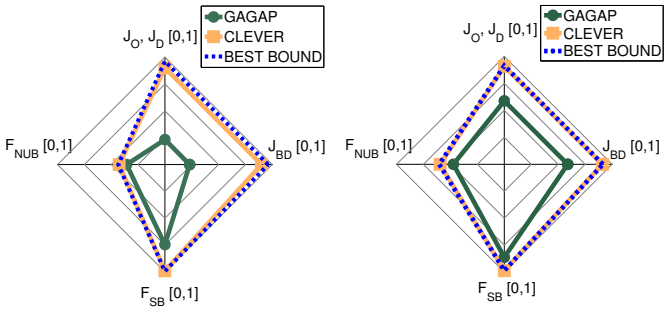


Fig. 7. Delay fairness index J_D vs. B_{MTO} for CLEVER and BEST BOUND.



(a) $B_{MTO} = 2$ Mhz (b) $B_{MTO} = 5$ Mhz

Fig. 8. Radar plots of CLEVER, GAGAP, and BEST BOUND considering: fraction of served bandwidth F_{SB}, fraction of not used bandwidth F_{NUB}, fairness of the bandwidth difference J_{BD}, fairness of delay J_D (CLEVER, BEST BOUND) or outage J_O (GAGAP).

(see Tab. 6), and ii) the actual bandwidth requested by each user is always the one computed toward the μeNB, which is in general much lower than the one computed toward the MeNB (due to the higher CQI values of the former compared to the latter). Nevertheless, our results suggest that CLEVER, with a proper setting of the B_{MTO} parameter, is able to trade between the reduction of the users delay (i.e., high values of B_{MTO}) and the limitation of the amount of MeNB resources used by offloaded users (i.e., low values of B_{MTO}).

Next, we compute from Eq. (18) the average fairness index on the delay J_D for CLEVER and BEST BOUND. Fig. 7 reports the obtained average values and confidence intervals of J_D for different values of B_{MTO}. Interestingly, the values of fairness are more than 0.8 in all the cases, thus suggesting that both the two solutions are also very effective in distributing the delay events across the set of users and the set of eNBs. Moreover, we can note that CLEVER is always very close to the BEST BOUND, even when B_{MTO} is decreased.

To give more insight, Fig. 8 reports the radar plots of CLEVER, BEST BOUND and GAGAP by considering the following metrics: i) average delay fairness index J_D from Eq. (18) (CLEVER and BEST BOUND), average outage fairness index J_O as in Eq. (19) (GAGAP), average fairness index of the difference in bandwidth J_{BD} as in Eq. (20), iii) fraction of not used bandwidth F_{NUB} as in Eq. (16), iv) fraction of served bandwidth F_{SB} as in Eq. (17). Note that all these metrics range from 1, corresponding to better values, to 0, corresponding to worse ones. As a result, the solution spanning the largest coverage over the metrics area is the best one. Fig. 8(a) reports the performance when $B_{MTO} = 2$ [Mhz]. Interestingly, we can see that CLEVER is pretty close to BEST BOUND, while GAGAP performs consistently worse. The low performance of GAGAP in this case is due to multiple reasons: i) the low value of B_{MTO} imposes to leave different users in outage, ii) users in outage are not served at all, thus reducing F_{SB}, ii) the users in outage tend to be always the same across the set of chunks, thus dramatically reducing the fairness indexes J_O and J_{BD}. On the contrary, CLEVER is able to wisely manage the users and the bandwidth assignment, by guaranteeing that all users receive a fair and effective allocation of the available bandwidth. Fig. 8(b) reports the performance when $B_{MTO} = 5$ [Mhz]. In this case, the performance of GAGAP is better compared to the B_{MTO} =2 [MHz] case, but still pretty far from CLEVER, which is instead almost the same of BEST BOUND.

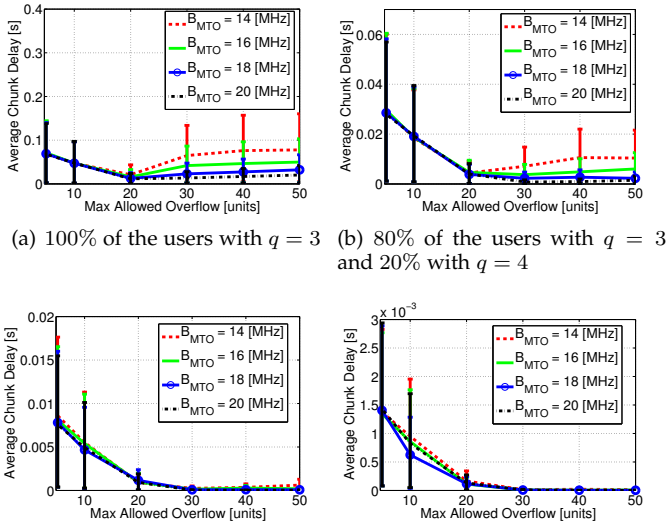
5.3 Results from the real city scenario

In this section, we present and discuss the system performance behavior results that we have obtained by studying the CLEVER scheme for the real city scenario. In this scenario, we consider two groups of users, each of them streaming at quality $q = 3$ and $q = 4$, respectively. The groups represent a percentage $p_{HQ} = p$ and $p_{LQ} = (1 - p)$ of the N users. Our experiments are repeated for 100 runs in generating the users positions and the video chunks.

Firstly, we study the performance of the scheme for different values of the maximum number of users $N_{MAX}^{(m)}, m = 0, \cdots M-1$ that can be offloaded from a generic μeNB to the MeNB. Since, in this case, $N_{MAX}^{(m)}$ can be lower than the number of users per μeNB $N_k^{(m)}$, we replace the line 13 of Alg. 1 with the MAD routine, which is invoked for the set of users not offloaded to the MeNB. Figs.9(a)-(d) plot the average delay per chunk $\overline{\delta}$ vs $N_{MAX}^{(m)}, m = 0, \cdots M-1$; within each sub-figure we plot four curves of the average delay per chunk $\overline{\delta}$ observed for different values of B_{MTO}. The figure reports also the confidence intervals obtained over the different runs (by assuming a 95% of confidence level).

Figs.9(a)-(d) are obtained for $p = 1, p = 0.80, p = 0.60, p = 0.40$ and therefore correspond to decreasing eNBs load. This is reflected into different magnitude orders of the per chunk average delay. Fig. 9(a) is obtained when $p = 1$, i.e., 100% of the users are streaming at the video quality

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2774298, IEEE Transactions on Mobile Computing



(a) 100% of the users with $q = 3$ (b) 80% of the users with $q = 3$ and 20% with $q = 4$

(c) 60% of the users with $q = 3$ and 40% at $q = 4$ (d) 40% of the users with $q = 3$ and 60% with $q = 4$

Fig. 9. Average delay per chunk $\bar{\delta}$ vs. the maximum number of offloaded users $N_{MAX}^{(m)}$ for different values of the B_{MTO} and different video qualities q.

$q = 3$. We recognize an optimal maximum number of users that should be allowed to overflow. This number is equal to 20. Beyond this value, the average delay value is noted to not continue any more to be reduced but rather to start increasing.

The reason is that as far as the maximum number of offloaded users' chunks increases we observe a twofold effect, namely: i) a reduction of the delay incurred by chunks streamed to μeNBs users and ii) an increase of the delay incurred by offloaded chunks streamed by the MeNB. Thereby, after a certain value of $N_{MAX}^{(m)}$, the beneficial effect on the μeNB is overtaken by the detrimental effect on the MeNB. Interestingly, the presence of this minimum level is mitigated when B_{MTO} is increased. In addition, Fig. 9(b) reports the case when $p = 0.8$, i.e., 80% of the users download their videos at quality $q = 3$, while 20% download at quality $q = 4$. From Fig. 9(a)-9(b), we can note that, when the average encoding rate is decreased, also the average delay tends to be increased. This is due to the fact that the average bandwidth demand level is decreased. As a result, it is easier for CLEVER to better serve the users. In addition to this, Fig. 9(b) shows also a variation in the minimum delay achieved by the different curves when B_{MTO} is greater than 14 [Mhz]. More in depth, the optimal values of $N_{MAX}^{(m)}$ increases from 20 to 30. Fig. 9(c) is obtained by setting $p = 0.6$, i.e., 60% of users to download video streams at quality $q = 3$ and 40% of them to download video streams at quality $q = 4$; conversely, in Fig. 9(d), 40% of users stream at $q=3$, while 60% of users stream at $q=4$. In the latter two cases, the optimal value of $N_{MAX}^{(m)}$ is stabilized at around 30.

To elaborate further, we here highlight the connection between the streaming QoE and the allowed B_{MTO} using the CLEVER approach. In HAS, the QoE [29] is expressed in terms of i) visual quality, determined by the experienced average video playout PSNR (which is equal to the encoding PSNR due to the user of reliable protocols), ii) fluidity, in

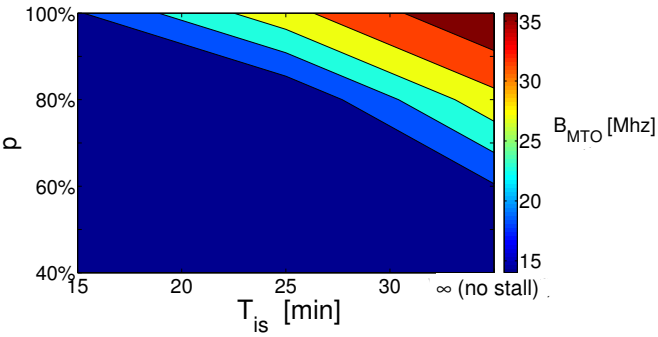


Fig. 10. Minimum B_{MTO} required to achieve a target T_{is} when different percentages $p_{HQ} = p, p_{LQ} = 1-p$ of HQ and LQ users, downloading videos at quality $q = 3$ and $q = 4$, respectively, are considered.

terms of reduced number and duration of playout interruptions (also named stalls) and iii) video smoothness, in terms of reduced number of rate switching. In the herein adopted experimental settings, the visual quality is summarized by the parameter $p = p_{HQ} = 1 - p_{LQ}$ which determines the average rate:

$$\overline{R} = p_{HQ} \cdot R_{HQ} + p_{LQ} \cdot R_{LQ}$$

as well as the average observed PSNR:

$$\overline{\text{PSNR}} = p_{HQ} \cdot \text{PSNR}_{HQ} + p_{LQ} \cdot \text{PSNR}_{LQ}$$

which in our settings equals to $\overline{R} = R_4 + p \cdot (R_3 - R_4)$ and $\overline{\text{PSNR}} = \text{PSNR}_4 + p \cdot (\text{PSNR}_3 - \text{PSNR}_4)$, respectively. As for the fluidity, this is clearly related to the buffer depletion rate, which is in turn affected by the average per-chunk delay $\bar{\delta}$. The average inter-stall time period T_{is}, i.e., the average time between two consecutive stalls, depends on multiple factors, including the playout time τ, the initial buffering (or rebuffering) time T_b and the average per chunk delay $\bar{\delta}$ during inter-stall time periods. For an initial buffering of T_b seconds, a stall occurs when the average accumulated delay equals the buffered video data duration, i.e., after a number K_s of played-out chunks that is approximated as: $K_s \bar{\delta} \approx T_b$, i.e., after an inter-stall time period T_{is} calculated as: $T_{is} = K_s \cdot \tau$. Therefore, as a rule of thumb we obtain:

$$T_{is} \cdot \bar{\delta} \approx \tau T_b \tag{21}$$

This approximation can be used to limit the acceptable maximum chunk delay value. For example, assuming that $T_{is} = 10\,[\text{min}] = 600\,[\text{s}]$, $T_b = 5\,[\text{s}]$, and $\tau = 2\,[\text{s}]$, we deduce $\bar{\delta} = 0.017\,[\text{s}]$.

As for the HAS smoothness, most of rate adaptation methods take into account the buffer state in order to perform rate switching. Thereby, the reduction of the average chunk delay is expected to a reduced (or at worst unaltered) number of rate switching, thanks to the improved (diminished) buffer depletion rate.

With these positions, we analyze the (minimum) B_{MTO} required to achieve a target interval within stalls T_{is} when $p_{HQ} = p$. Specifically, Fig. 10 shows the level curves of the (minimum) B_{MTO} required to achieve T_{is} for a given value of p (with $T_b = 5$ [s]). As expected, the (minimum) B_{MTO} increases with the average inter-stalls interval T_{is} and with the value of p (which corresponds to increase $\overline{R}$ and $\overline{\text{PSNR}}$).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2774298, IEEE Transactions on Mobile Computing

From Fig. 10 we recognize that when a given MeNB bandwidth level is available for the offloading (e.g., 20[MHz], light black in the plot), it can be spent to improve the users QoE either by setting a longer T_{is} level (i.e., increasing the video stream fluidity) or by setting a larger percentage p of users to employ a higher encoding bit-rate (i.e., by increasing the visual quality level). This paves the way to define different utility metrics for designing pricing policies and service admission procedures, which is left for future work.

6 Related Work

Mechanisms for resource allocation across heterogeneous networks have been studied in several papers. Specifically, in [30] and [31], the authors propose and analyze mechanisms to offload the MeNBs to small cells (e.g., to hot spots) to alleviate the loading of MeNBs and improve the reception quality (QoE) perceived by users. The authors investigate the problems caused by the different channel conditions, including co-channel interference, over the two different cell types. A comprehensive survey of data offloading techniques in cellular networks is presented in [32], while emphasis on the role played by packet schedulers in 4G systems is given in [33]. The architectural model adopted in this paper is similar to the one proposed in [34][35] where the HetNet is composed of an MeNB and several μeNBs. We use the case of an orthogonal deployment, i.e., where μeNBs are allocated a pool of subchannels (i.e., a frequency band) orthogonal to the set of subchannels used for MeNB operation. Like in those works, we schedule the transmissions with a simple eNBs coordination by having an eNB that can be transmit or not transmit at all. A key difference is that our rule for assigning users to the different eNBs available in the system depends also on the application at hand (HAS in or case), which may have strict constraints as for the QoE. Therefore, we show that, besides the benefits detailed by [34][35] as for the use of a suitable Hetnets scheduling, the advantage in case of video streaming is much more significant. Finally, the paper [36] presents an optimization framework to evaluate the performance of radio resource strategies in uplink for HetNets, with regard to both the interferences and a proper power control algorithm.

Multimedia transmission, and specifically video streaming, is becoming the dominant source of traffic in cellular systems. In particular, the streaming of video contents is expected to consume up to 80% of global IP data traffic over the next several years [37]. In this context, solutions which are able to jointly optimize video transmission and network resource allocation are of fundamental importance. For example, the goal of [38] is to maximize the total reception quality of a limited number of video stream flows conducted across a single-cell network with mixed voice and video users. The study in [39] presents a flow management framework that performs joint optimal scheduling of resources across multiple adaptive video streaming flows. The authors propose an allocation algorithm that balances between stability of a user's bit-rate and efficient resource utilization of the base station. The paper [40] provides a mechanism (called GTube) for an efficient DASH streaming based on a selection of the video quality on the basis of the future geographical positions of the end users and their trajectories. This information is collected and processed by a server to help the client in making quality adaptation decisions. In our case instead, the geographical positions of the end users are implicitly used (through the CQI) to select the best serving eNB by combining them with the application layer information (i.e., the chunk size). Finally, detailed discussion on the challenges in providing adaptive media streaming to mobile devices is provided in [41].

7 Conclusions

We have investigated the problem of providing a video streaming service to users in an heterogeneous cellular network composed of μeNBs and MeNBs, by targeting the minimization of the chunk delay. After optimally formulating the MTD problem, we have presented CLEVER, an algorithm explicitly tailored to the reduction of the chunk delay, while being able to control the number of users offloaded to the MeNB. CLEVER makes use of information retrieved from the application layer (residing at the video server) as well as of status data residing at the physical and data-link layers. We have then compared CLEVER against a set of reference algorithms (namely NO OFFLOADING, BEST BOUND and GAGAP), by considering two reference scenarios and different performance metrics. Our results confirm that CLEVER limits the average chunk delay experienced by users, by being also able to achieve high fairness indexes. In addition, the performance of CLEVER is much better than NO OFFLOADING and GAGAP, while being very close to the BEST BOUND. Moreover, we have shown that CLEVER is able to exploit the μeNBs and MeNBs resources in terms of used bandwidth, while limiting the maximum number of users to be offloaded $N_{MAX}^{(m)}$. Finally, we have shown that our solution can be tuned to trade between the amount of reserved bandwidth on the eNBs and the QoE perceived by users.

Future work will include the study of coordination and optimization of the resource allocation schemes while including also mechanisms that can incorporate offloading of a portion of a video chunk. The approach then involves the application of the joint allocation scheme over the μeNBs while offloading to the MeNB only the residual bandwidth that is needed following the execution of an optimal bandwidth allocation process over the μeNB. Such a process is expected to improve the utilization of μeNBs resources.
Moreover, our work can be extended by accounting for the different backhauls delays incurred by each chunk. Finally, our study can be used for the development of a software-defined resource allocation entity, i.e., for systems where the bandwidth of heterogeneous cells is not a priori set, but is rather transferred between cells as it dynamically tracks traffic processes and service requirements.

References

[1] S. Colonnese, V. Salvatore, L. Chiaraviglio, and F. Cuomo, "Dynamic and Cooperative Mobile Video Streaming Across Heterogeneous Cellular Networks," in *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, June 2016, pp. 1–8.

[2] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. Soong, and J. C. Zhang, "What will 5G be?" *Selected Areas in Communications, IEEE Journal on*, vol. 32, no. 6, pp. 1065–1082, 2014.

[3] A. Khandekar, N. Bhushan, J. Tingfang, and V. Vanghi, "LTE-advanced: Heterogeneous networks," in *Wireless Conference (EW), 2010 European*. IEEE, 2010, pp. 978–982.

[4] A. Damnjanovic, J. Montojo, Y. Wei, T. Ji, T. Luo, M. Vajapeyam, T. Yoo, O. Song, and D. Malladi, "A survey on 3GPP heterogeneous networks," *Wireless Communications, IEEE*, vol. 18, no. 3, pp. 10–21, 2011.

[5] V. Jungnickel, K. Manolakis, W. Zirwas, B. Panzner, V. Braun, M. Lossow, M. Sternad, R. Apelfrojd, and T. Svensson, "The role of small cells, coordinated multipoint, and massive MIMO in 5G," *Communications Magazine, IEEE*, vol. 52, no. 5, pp. 44–51, 2014.

[6] A. Gudipati, D. Perry, L. E. Li, and S. Katti, "SoftRAN: Software defined radio access network," in *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*. ACM, 2013, pp. 25–30.

[7] I. Rubin, S. Colonnese, F. Cuomo, F. Calanca, and T. Melodia, "Mobile HTTP-based streaming using flexible LTE base station control," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2015 IEEE 16th International Symposium on a*, June 2015, pp. 1–9.

[8] S. Colonnese, F. Cuomo, T. Melodia, and R. Guida, "Cloud-assisted buffer management for http-based mobilevideo streaming," in *Proceedings of the 10th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks*, ser. PE-WASUN '13, 2013, pp. 1–8.

[9] D. Lee, H. Seo, B. Clerckx, E. Hardouin, D. Mazzarese, S. Nagata, and K. Sayana, "Coordinated multipoint transmission and reception in lte-advanced: deployment scenarios and operational challenges," *Communications Magazine, IEEE*, vol. 50, no. 2, pp. 148–155, February 2012.

[10] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hossfeld, and P. Tran-Gia, "A Survey on Quality of Experience of HTTP Adaptive Streaming," *Communications Surveys Tutorials, IEEE*, vol. PP, no. 99, pp. 1–1, 2014.

[11] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive," in *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '12, 2012, pp. 97–108.

[12] S. Colonnese, S. Russo, F. Cuomo, T. Melodia, and I. Rubin, "Timely Delivery Versus Bandwidth Allocation for DASH-Based Video Streaming Over LTE," *IEEE Communications Letters*, vol. 20, no. 3, pp. 586–589, March 2016.

[13] R. M. Nauss, "Solving the generalized assignment problem: An optimizing and heuristic approach," *INFORMS Journal on Computing*, vol. 15, no. 3, pp. 249–266, 2003.

[14] S. Colonnese, P. Frossard, S. Rinauro, L. Rossi, and G. Scarano, "Joint source and sending rate modeling in adaptive video streaming," *Signal Processing: Image Communication*, vol. 28, no. 5, pp. 403 – 416, 2013.

[15] S. Colonnese, F. Cuomo, T. Melodia, and I. Rubin, "A cross-layer bandwidth allocation scheme for http-based video streaming in lte cellular networks," *IEEE Communications Letters*, vol. 21, no. 2, pp. 386–389, Feb 2017.

[16] D. Bethanabhotla, G. Caire, and M. J. Neely, "Wiflix: Adaptive video streaming in massive mu-mimo wireless networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 6, pp. 4088–4103, June 2016.

[17] S. Pudlewski, N. Cen, Z. Guan, and T. Melodia, "Video transmission over lossy wireless networks: A cross-layer perspective," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 1, pp. 6–21, Feb 2015.

[18] M. Zhao, X. Gong, J. Liang, W. Wang, X. Que, and S. Cheng, "Qoe-driven cross-layer optimization for wireless dynamic adaptive streaming of scalable videos over http," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 3, pp. 451–465, March 2015.

[19] G. Nigam, P. Minero, and M. Haenggi, "Coordinated multipoint joint transmission in heterogeneous networks," *IEEE Transactions on Communications*, vol. 62, no. 11, pp. 4134–4146, Nov 2014.

[20] J. Walfisch and H. L. Bertoni, "A theoretical model of UHF propagation in urban environments," vol. 36, no. 12, Dec 1988, pp. 1788–1796.

[21] F. Ikegami, S. Yoshida, T. Takeuchi, and M. Umehira, "Propagation factors controlling mean field strength on urban streets," *IEEE Transactions on Antennas and Propagation*, vol. 32, no. 8, pp. 822–829, 1984.

[22] C. Mehlfuhrer, M. Wrulich, J. Ikuno, D. Bosanska, and M. Rupp, "Simulating the Long Term Evolution physical layer," in *Signal Processing Conference, 2009 17th European*, Aug 2009, pp. 1471–1478.

[23] J. C. Ikuno, M. Wrulich, and M. Rupp, "System Level Simulation of LTE Networks," in *Vehicular Technology Conference (VTC 2010-Spring), 2010 IEEE 71st*, May 2010, pp. 1–5.

[24] P. Vieira, P. Queluz, and A. Rodrigues, "LTE spectral efficiency using spatial multiplexing MIMO for macro-cells," in *Signal Processing and Communication Systems, 2008. ICSPCS 2008. 2nd International Conference on*, Dec 2008, pp. 1–6.

[25] Emilia Romagna Electromagnetic Map . [Online]. Available: http://www.arpae.it/cem/webcem/bologna/

[26] "Video Trace Files and Statistics," 2014. [Online]. Available: http://trace.eas.asu.edu/videotraces2/h265/

[27] P. Seeling and M. Reisslein, "Video transport evaluation with h.264 video traces," *IEEE Communications Surveys Tutorials*, vol. 14, no. 4, pp. 1142–1165, Fourth 2012.

[28] H. Kellerer, U. Pferschy, and D. Pisinger, "Knapsack problems," *Springer, Berlin*, 2003.

[29] J. M. Batalla, P. Krawiec, A. Beben, P. Wisniewski, and A. Chydzinski, "Adaptive video streaming: Rate and buffer on the track of minimum rebuffering," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 8, pp. 2154–2167, Aug 2016.

[30] S. Singh and J. G. Andrews, "Joint resource partitioning and offloading in heterogeneous cellular networks," *IEEE Transactions on Wireless Communications*, vol. 13, no. 2, pp. 888–901, February 2014.

[31] X. Duan, A. M. Akhtar, and X. Wang, "Software-defined networking-based resource management: data offloading with load balancing in 5G HetNet," *EURASIP Journal on Wireless Communications and Networking*, vol. 2015, no. 1, pp. 1–13, 2015.

[32] F. Rebecchi, M. D. de Amorim, V. Conan, A. Passarella, R. Bruno, and M. Conti, "Data offloading techniques in cellular networks: A survey," *IEEE Communications Surveys Tutorials*, vol. 17, no. 2, pp. 580–603, 2015.

[33] F. Capozzi, G. Piro, L. A. Grieco, G. Boggia, and P. Camarda, "Downlink packet scheduling in lte cellular networks: Key design issues and a survey," *IEEE Communications Surveys Tutorials*, vol. 15, no. 2, pp. 678–700, Second 2013.

[34] J. Ghimire and C. Rosenberg, "Resource allocation, transmission coordination and user association in heterogeneous networks: A flow-based unified approach," *IEEE Transactions on Wireless Communications*, vol. 12, no. 3, pp. 1340–1351, March 2013.

[35] D. Fooladivanda and C. Rosenberg, "Joint resource allocation and user association for heterogeneous wireless cellular networks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 1, pp. 248–257, January 2013.

[36] J. P. Muñoz-Gea, R. Aparicio-Pardo, H. Wehbe, G. Simon, and L. Nuaymi, "Optimization framework for uplink video transmission in hetnets," in *Proceedings of the 6th ACM Mobile Video Workshop, MoVid 2014, Singapore, March 19, 2014*, 2014, pp. 6:1–6:6. [Online]. Available: http://doi.acm.org/10.1145/2579465.2579467

[37] "Cisco white paper, visual networking index: Global mobile data traffic forecast update, 2012-2017," Feb. 2013.

[38] J. Huang, Z. Li, M. Chiang, and A. K. Katsaggelos, "Joint source adaptation and resource allocation for multi-user wireless video streaming," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 5, pp. 582–595, May 2008.

[39] J. Chen, R. Mahindra, M. A. Khojastepour, S. Rangarajan, and M. Chiang, "A scheduling framework for adaptive video delivery over cellular networks," in *Proceedings of the 19th Annual International Conference on Mobile Computing and Networking,Miami, Florida, USA*, 2013, pp. 389–400.

[40] J. Hao, R. Zimmermann, and H. Ma, "Gtube: Geo-predictive video streaming over http in mobile environments," in *Proceedings of the 5th ACM Multimedia Systems Conference*, ser. MMSys '14. New York, NY, USA: ACM, 2014, pp. 259–270. [Online]. Available: http://doi.acm.org/10.1145/2557642.2557647

[41] K. Evensen, T. Kupka, H. Riiser, P. Ni, R. Eg, C. Griwodz, and P. Halvorsen, "Adaptive media streaming to mobile devices: Challenges, enhancements, and recommendations," *Adv. MultiMedia*, vol. 2014, pp. 10:10–10:10, Jan. 2014. [Online]. Available: http://dx.doi.org/10.1155/2014/805852

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2774298, IEEE Transactions on Mobile Computing

Algorithm 2 Pseudo-Code of the BEST BOUND algorithm on the μeNBs

Input: k, $B_k^{(m,i)}$, $B_{MAX}^{(m)} m = 0, \cdots M-1, i \in \mathcal{S}_k^{(m)}$
Output: $\tilde{B}_k^{(m,i)}, m = 0, \cdots M-1, i \in \mathcal{S}_k^{(m)}$
1: **for** $m = 0...M-1$ **do**
2: **MAD** to **compute** $\tilde{B}_k^{(m,i)}$:
3: **define** $\Gamma = B_{MAX}^{(m)}$, $\Omega = \mathcal{S}_k^{(m)}$,
4: $\mathcal{T} = \{i \in \mathcal{S}_k^{(m)}, i \; s.t \quad B_k^{(m,i)} \leq (B_{MAX}^{(m)}) \cdot \sqrt{B}_k^{(m,i)} / \sum_{l \in \mathcal{S}_k^{(m)}} \sqrt{B}_k^{(m,l)}\}$
5: **while** $\mathcal{T} \neq \emptyset$ **do**
6: $\tilde{B}_k^{(m,i)} = B_k^{(m,i)}, i \in \mathcal{T}$;
7: $\Gamma = \Gamma - \sum_{i \in \mathcal{T}} \tilde{B}_k^{(m,i)}$;
8: $\Omega = \Omega \setminus \mathcal{T}$
9: $\mathcal{T} = \{i \in \Omega, i \; s.t \; B_k^{(m,i)} \leq \Gamma \cdot \sqrt{B}_k^{(m,i)} / \sum_{l \in \Omega} \sqrt{B}_k^{(m,l)}\}$
10: **end while**
11: $\tilde{B}_k^{(m,i)} = \Gamma \cdot \sqrt{B}_k^{(m,i)} / \sum_{l \in \Omega} \sqrt{B}_k^{(m,l)}, i \in \Omega$;
12: **end for**

Algorithm 3 Pseudo-Code of the GAGAP algorithm

Input: k, $\mathcal{M}$, $B_k^{(m,i)}$, $B_{MAX}^{(m)}$
Output: $x_k^{(m,i)}$ $\tilde{B}_k^{(m,i)}$
1: $B_{RES}^{(m)} = B_{MAX}^{(m)} \quad \forall m \in \mathcal{M}$;
2: $x_k^{(m,i)} = 0 \quad \forall m \in \mathcal{M}, \forall i \in \mathcal{S}$;
3: **for** $m = 0...M$ **do**
4: $\mathcal{S}'$=sort($B_k^{(m,i)}$,ascend)
5: **for** $i \in \mathcal{S}'$ **do**
6: **if** $\sum_{m' \in \mathcal{M}, m' \neq m} x_k^{(m',i)} == 0$ && $B_k^{(m,i)} > 0$ **then**
7: **if** $B_{RES}^{(m)} - B_k^{(m,i)} > 0$ **then**
8: $x_k^{(m,i)} = 1$;
9: $\tilde{B}_k^{(m,i)} = B_k^{(m,i)}$;
10: $B_{RES}^{m} = B_{RES}^{m} - B_k^{(m,i)}$;
11: **end if**
12: **end if**
13: **end for**
14: **end for**

APPENDIX A
BEST BOUND DESCRIPTION

The BEST BOUND algorithm operates by applying the MAD algorithm on each and every μeNB by allocating the overall bandwidth B_{MTO}+$B_{\mu eNB}$ to the μeNB users. Alg. 2 reports the BEST BOUND pseudo-code. In particular, this solution is able to exploit the full available bandwidth $B_{MAX}^{(m)} = B_{MTO}$+$B_{\mu eNB}$ as in Tab. 6 without constraints. Therefore, it can perform even better than CLEVER, which instead operates separately on the μeNBs and the MeNB. Besides, in evaluating the bandwidth requested, we consider the one requested to the μeNBs: this is equivalent to assume the B_{MTO} to be actually available at the μeNB. Since for μeNB users $B_k^{(m,i)} \leq B_k^{(M,i)} m = 0, \cdots M-1, i \in \mathcal{S}_k^{(m)}$, the users require an overall bandwidth lower than the one that would be requested to the MeNB in case of offloading.

As for the BEST BOUND time complexity, apart from computing the requested bandwidth $B_k^{(m,i)}$ for each user and each eNB at a complexity $O(M \times N)$, applies the MAD algorithm on each μeNB. As already discussed, the recursive routine is rarely run, so that the amount of served bandwidth is straightforwardly computed for the μeNB users. As a result, the overall complexity of the BEST BOUND algorithm is approximated as $\approx O(M \times N)$. We then consider the space complexity of the BEST BOUND. Storing the requested bandwidth $B_k^{(m,i)}$ and the assigned bandwidth $\widetilde{B}_k^{(m,i)}$ in two matrices requires to store up most $(M+1) \times N$ elements for each chunk index k.

APPENDIX B
GAGAP ALGORITHM DESCRIPTION

In this section, we provide a brief overview of the GAGAP algorithm [28], which is a greedy heuristic to solve the Generalized Assignment Problem. In our case, we have adapted GAGAP to solve the problem for each chunk index k. More in detail, the main idea of the heuristic is to iteratively assign the users to the eNBs, by serving each user with the requested amount of bandwidth, until there is an amount of residual bandwidth available on the eNBs. Initially, the amount of residual bandwidth available on the eNB is set the maximum one, i.e., $B_{RES}^{(m)} = B_{MAX}^{(m)}$ (line 1). Moreover, the association of users to the eNB is also initialized to 0 values (line 2). The algorithm then iterates over the set of eNBs, from the μeNB to the MeNB (line 3). For each eNB m, the users are sorted by increasing values of $B_k^{(m,i)}$ (line 4). The sorted users are stored in the set $\mathcal{S}'$. Then, for each user $i \in \mathcal{S}'$, if the user has not been previously assigned to another eNB and $B_k^{(m,i)}$ is larger than 0 (line 6), the algorithm tries to associate the user to m. Specifically, if there is enough bandwidth on the m-th eNB (line 7), the user is served by m (line 8). Hence, the amount of served bandwidth is set equal to the requested one (line 9), and the residual available bandwidth on m is updated (line 10). Otherwise, the user is not served by the current eNB, and has to be offloaded to the following eNB. The procedure ends when all the eNBs are analyzed. Note that, since GAGAP always tries to fully satisfy the bandwidth requested by each user, the delay $\delta_k^{(m,i)}$ is equal to zero for the served users. Clearly, if the amount of bandwidth is not enough to serve all the users, some of them will be in outage condition, i.e., $\tilde{B}_k^{(m,i)} = 0$ and $x_k^{(m,i)} = 0$. Such users will be not be served by any eNB, resulting in a loss of data equal to $B_k^{(m,i)}$ for each of them.

Focusing on the time complexity of GAGAP, the sorting of the users requires $\mathcal{O}(N \log(N))$. Then, the sorting has to be repeated for all the eNBs $(M+1)$, resulting in a total complexity $\mathcal{O}(M \times N \log(N))$. Focusing then on space complexity, GAGAP requires to store the assignment matrix $x_k^{(m,i)}$, which has size $K \times (M+1) \times N$. In addition, a temporary array of N elements is required to store the sorted users. Moreover, the size of the matrices $\tilde{B}_k^{(m,i)}$ and $B_k^{(m,i)}$ is $K \times (M+1) \times N$. Finally, the arrays $B_{RES}^{(m)}$ and $B_{MAX}^{(m)}$ have $(M+1)$ elements. The total space complexity is then $3 \times K \times (M+1) \times N + N + 2 \times (M+1)$.